DEFENCE AND CIVIL INSTITUTE
OF ENVIRONMENTAL MEDICINE

Canada

November, 1986

DCIEM No. 86-R-47

Systems Design Description of A Physiological Signals Management System (PSMS)

C.E. Beeton
W.D. Fraser

Defence and Civil Institute of Environmental Medicine
1133 Sheppard Avenue West, P.O. Box 2000
Downsview, Ontario M3M 3B9

DEPARTMENT OF NATIONAL DEFENCE - CANADA

Table of Contents

Abstract i

Introduction 1

Needs Analysis 1

Problem Definition 3

Definitions 4

Inputs to the System 5

Data File Input 7

Selection of Data Files 7

Application 11

User Interface 11

The Complete System 14

Further Development 14

Conclusions and Recommendations 15

Acknowledgements 15

Figures 16

Annex A: SDBMS 19

Annex B: Analysis of Force Platform Data 20

ABSTRACT

This report describes the analysis of a possible computer-based system to manage the large number of extensive computer data files created during human physiological experiments. The analysis has been developed specifically to address the requirements of a particular laboratory, the Aerospace Group (ASG) at DCIEM, but is applicable to any experimental laboratory group collecting large amounts of physiological, engineering, or similar time series data.

This report represents the first step in the design of a Physiological Signals Management System (PSMS). Following Systems Design methodology, a need in the research environment is analyzed, and a specific problem definition developed. A detailed consideration of the system components, including the human interface, then follows. Some recommendations about the structure of the system are made. A feasibility study and detailed technical design will be required before such a system could be implemented.

INTRODUCTION

The Defence and Civil Institute of Environmental Medicine (DCIEM) is a research facility of the Canadian Department of National Defence (DND) whose primary role is the investigation of the performance of humans in hostile environments such as high altitude, diving, or extreme cold and heat.

Numerous complex experiments are performed on human subjects and large amounts of physiological data are collected and stored on computer tapes and disks. The scope of the data is tremendous, and the organizational problems of analyzing it have become more time-consuming than the data collection itself.

The purpose of this systems design study was to develop a framework of a computer-based system to manage the data files created during physiological experiments. The system design has been developed specifically for the Aerospace Group (ASG) at DCIEM, but is applicable to any experimental group collecting large amounts of physiological or engineering data.

NEEDS ANALYSIS

Experiments conducted at DCIEM monitor physiological parameters such as cardiac output, brain activity, and arterial hemoglobin saturation level etc. Many of these signals are recorded as wave-forms. During a single run of an experiment, large volumes of raw signal data may be collected and stored on disk, magnetic tape, or paper printouts.

There are two basic methods of physiological data collection and analysis. The first methods records the changes in signals over time on "strip charts". Figure 1 illustrates a typical section of a strip chart page, showing blood pressure, ECG, and stroke volume. A researcher examines the chart and identifies important points such as peaks and valleys. Calculations are made manually, and the values are manually entered into a computer for processing by an analysis package. The process will take months, if a substantial amount of data is collected.

The second method is to use a computer to sample the monitoring instruments regularly and record the data in a raw data file. Computer algorithms are developed to identify significant points on the digitized wave form, and calculate values from these points. These calculations may be done during or after data acquisition.

The computer can perform most of the functions which, in the past, were done manually. The use of a computer for data collection increases efficiency and reduces the amount of time spent preparing the data for statistical analysis. Another benefit of computer data collection systems is the consistency in analysis which is difficult for a human to achieve. The major problem with the computer is the creation of a large number of computer data files on tapes and disks.

The data is often spread among several computer files, one for each monitored signal. Additional processed data files may be created by the data acquisition system, or

later by a more refined analysis of the raw signals. Currently, the relationships among the various files, runs, and experiments must be tracked manually with the possibility of human error, at the worst, and a very time consuming task, at best. As well, to prepare raw data for analysis, the format of a data file must be changed to conform to what the analysis package expects. Often, data from several files must be combined. Thus the files must be edited and re-arranged, and headers must be added, all by single key stroke entries or by small *ad hoc* editing programs. As an example, see Annex B for a description of the steps taken to prepare force platform data for analysis.

It often takes many weeks, or even months, for the data generated by an experiment to be fully analyzed. Due to the quantity of the data, it is difficult and time-consuming to examine several different experiments together. Thus little cross-referencing is done between related experiments.

Several programs are commonly used for data analysis at DCIEM, specifically in the Aerospace Group. It is anticipated that additional programs will be added in the future.

The program most often used is the BMDP (Biomedical computer programs) software [4]. This is a statistical analysis package which consists of many modules performing different analyses (t-test, variance, regression , etc.).

Also used are small analysis programs developed for DCIEM. For example, one program reads the raw data file created during force platform experiments and calculates approximately 100 parameters describing the postural stability of human subjects. Other programs are under development to perform standard ECG waveform analysis, detect peaks and valleys on waveforms, and so on. The calculated values from many experimental runs are passed to the BMDP package for statistical analysis.

Other programs which will be used may include GPAC, a graphics package [15]; ILS, an interactive package which can perform some calculations on raw data [17]; and S, an interpretive language for data analysis [3]. Possibilities for the future include RS/1, which contains some statistics, graphing, and analysis capabilities [2]; SAS, another statistics package [1]; and similar commercial packages. Smaller programs may be written in C as they are needed.

The lengthy data file manipulation and analysis process is undesirable. A great deal of time is spent by scientists and technicians organizing the many files and converting them into a form suitable for computer analysis. This work is inefficient and nonproductive. Since it takes so long for a single experiment to be analyzed, it is usually not feasible to compare results with other similar experiments in any comprehensive manner. However, such comparisons could yield valuable insight in some circumstances.

It has, therefore, been recognized that a more efficient and effective method of data management should be developed. A system is needed to help with the organization, management, and analysis of complex physiological signals. This problem has been addressed in this report. The needs have been defined more precisely, and some

suggestions have been proposed for the structure of the desired system.

PROBLEM DEFINITION

The system must be able to manipulate a number of different types of computer files to prepare data for statistical and graphical processing. The required functions of the system are summarized in the following points. The system must:

1. Organize the raw and processed data files created in the experiments. That is, store the file names and locations, and the relationships between the two types of files.
2. Allow a user to choose selected files (based on subject, experiment, type of signal, etc.), then locate the selected files and provide information about them.
3. Examine the experiment log files and notify the user of any extraordinary events (eg. run aborted).
4. Allow within-subject, within-experiment, cross-subject, and cross-experiment analyses, and within-signal and cross-signal analyses.
5. Allow the user to select files based on data in the files (especially processed data).
6. Prepare the selected files for statistical or graphical processing by various packages (ILS, S, BMDP), or for other application programs as specified by the user.
7. Provide a means to add new files into the system easily.
8. Allow new application software to be easily added into the system.

The system must be very flexible. The experiments are continually changing, and the format and contents of the data files (raw and processed) are not fixed. As the experiments change so will the application and analysis programs. New methods of analysis may be developed in the future so that old data may be re-analyzed. No limitations should be imposed on the structure of the data files or on the choice of application program.

The PSMS must be easy to use. The user should not be required to remember complex file names, numbers, or commands.

The system is to be designed specifically for physiological experiment data. The data files are always classifiable by subject and experiment, although they may contain any type of information.

Existing computer facilities (hardware and software) must be used in the design of the system. Programming is to be done in C and the UNIX shell language. A commercial data base such as INGRES is to be used as the data base. The system must be compatible with existing application software (BMDP, S, GPAC on the UNIX operating system, ILS on the TSX/PLUS and RT-11 operating systems), as well as with the operating system of the data acquisition systems (TSX/PLUS and RT-11).

Many laboratory data analysis packages have been developed by various institutions: LAOS [18]; ANDIRON [16] for cardiovascular data; ISDMS [5] for engineering and scientific time series data; GREEN/REGREN [14] for neural evoked potentials; etc. Commercial systems which are designed for signal and data processing are also available, such as ILS [17] and RS/1 [2]. These systems usually provide graphics, editing, and analysis of the data, thus satisfying some of the PSMS requirements.

Among the more sophisticated of the data analysis packages is the INEL Scientific Data Management System (ISDMS) [5]. This system was developed by a group at the Idaho National Engineering Laboratory (INEL), U.S. Department of Energy. The ISDMS is an integrated system which provides for the organization, processing, control, and retrieval of scientific data. The system arose out of the confusion caused by the different data structures and acquisition systems in use at INEL.

A number of modules which access the data files are available to the user through a command language. These include engineering analysis, data reformatting, and graphics. Programmers have access to utilities so that they can develop new applications. The system environment (the data at the core, surrounded by application modules and the user interface) is very similar to that proposed for the PSMS (see Section 10.).

This system, like most of the others listed above, has good analysis capabilities, but requires the user to keep track of what file comes from which run of which experiment. Compilation of data from different experiments or files is not straightforward. None of the systems form a complete integrated information organization and analysis system. The PSMS would provide these capabilities, and can thus make the researcher's job easier.

A similar large scale, high level data base system for spatial data is currently being developed at DCIEM. The system, Spatial Data Base Management System (SDBMS), is described in Annex A. Some of the research results about user interfaces and data structures may be applied to the PSMS (see for example Section 9).

DEFINITIONS

Before proceeding any further, some clarifications must be made about the terminology used in this report.

A *Raw file* refers to the binary file produced by the analogue to digital (A/D) converter when the experiment is run. It contains digitized signal data. A *Processed file* refers to any other kind of data file. Calculations performed using raw files produce processed files. Processed files can also be produced with a text editor, if a few well-defined values are needed (eg. blood chemistry data or Doppler bubble scores). Processed files are usually ASCII files.

The terminology related to runs and experiments is often confused. For the PSMS, a precise definition is needed, as organization of files is based on this distinction.

A *run* is created for each subject involved in a distinct set of conditions. If four subjects are involved in an experiment, then the data files are split between four runs, one for each subject. A trial is defined as single data collection epoch where a particular change in the subjects environment or action is required. There may be several trials within an experiment.

An *experiment* is a group of similar runs. For example, if four people participate in a dive, then each is associated with a *run*, while the dive itself is an *experiment*. Also, a high altitude positive pressure breathing experiment is made up of several runs, at different times, with different people. Conditions can vary within an experiment (if different breathing gas combinations are given in an experiment), or they can be constant (e.g., for all subjects). Each experiment also has a specific protocol associated with it.

INPUTS TO THE SYSTEM

The inputs to the system consist mainly of files containing several forms of data. Most of the files will be created on the TSX/PLUS and RT-11 systems by the data acquisition software currently under development. The various files are described in this section.

For any given experiment the raw data files contain digitized signals from many different instruments and experiments, such as:

- Electroencephalogram (EEG)
- Electrocardiogram (ECG)
- Force Platform
- Mass Spectrometer
- Gas Analyzers

The format of the raw data files differs according to both the signals being digitized and the acquisition software package being utilized. A file may contain values for one signal, or for many signals that have been multiplexed. Often, several raw data files will be associated with a single run (if more than one signal are being monitored).

Processed files contain information calculated from the raw data. Some processed files are produced directly by the data acquisition system, while others are produced later, by additional software packages (which may be run under either the UNIX or the TSX/RT-11 operating systems).

These files contain information such as:

- Cardiac ouput
- Stroke volume
- Heart rate
- ECG morphologies
- Systolic time intervals

Some processed files may be created with a text editor to introduce into the system data which was not collected through a data acquisition system. For example, data which was recorded with paper and pencil can be incorporated into the system in this way. Several processed data files may be associated with a single run of an experiment. It is also possible for many processed files to be developed from one raw data file. Raw data may be re-processed long after its collection, if new analysis methods are developed.

Two files identify the subject and experiment from which the data was obtained. The *Log File* contains the times and descriptions of events which occurred during data collection. Examples of events are a switch from one gas mixture to another, abort reasons, etc. Calibration signals and times are also stored in this file. The *Master File List*, created by the data acquisition package for each run, contains a list of all files (raw, processed, and text) related to that run.

Data collected in experiments by other research groups at DCIEM is stored in data bases and files created using different data acquisition systems. The largest of these systems is CANDID (Canadian Diving Data) [8]. CANDID is used internally at DCIEM to store data collected from diving experiments. This information can be retrieved by entering search criteria which are passed through CANDID to the INGRES data base. CANDID contains information describing the conditions of the experiment (date, dive type, maximum depth, bottom time, gas mixtures, divers, etc.), and the incidences of decompression sickness. A dive profile is stored separately as a UNIX file.

The nature of the data managed by the CANDID system is very different from that in the PSMS. CANDID deals with a set of well-defined integer values. The pneumatic computers (the closest CANDID comes to monitoring instruments) are sampled at intervals of several minutes (from one to 45, depending on the state of the dive).

The PSMS should eventually be able to interface with the data held in the CANDID system. With the help of a simple input module, data can be extracted from the CANDID data base and inserted into the PSMS. Since the CANDID data is not complete in terms of the PSMS, extra subject and experiment information will be prompted for.

One of the most important inputs to the system is provided by the researchers themselves, the users. The average user of the PSMS will be a defence scientist, likely without much formal training in computers. He or she will have little time to spend learning how to use the system, and may not use it regularly. Other more naive users, such as summer students, will also require access to the system.

DATA FILE INPUT

Most of the application packages (statistics, graphics, etc.) are installed on the UNIX systems at DCIEM. Thus, the files created by the data collection system under the TSX/RT-11 operating systems, must be translated into UNIX format.

In order to satisfy the required functions of the system, the files needed by the PSMS are: subject and experiment ID files, the log file, processed data files, and the master file list. Ideally, the raw files will also be transferred to a UNIX format. This will make on-line interfacing with the files much easier.

It may be necessary or desirable to convert the raw data files into a common format, so that the PSMS can access them. For example, the files may be converted to IPAC format, or perhaps to the common word addressable format (CWAF) used in the ISDMS system [5]. To save space, it might be desirable to reduce the raw data files into a compressed format. A detailed study of the nature of the signal data files is necessary to investigate these points.

To achieve the desired organization of files, and the search capabilities, a data base is needed. This data base would contain information such as subject and experiment names, date and time of experiment, and the names and locations of the raw and processed data files created in the experiment. A commercial data base could perform these functions. The raw data itself cannot be stored in a data base. This is because the format is too variable, and also because there is simply too much data (up to 100 megabytes of raw digitized data for one run). Thus, the data base can contain only a 'pointer' to the raw data file, that is, its name and location.

The addition of new experiments should be as automatic as possible (section 3), requiring only a command and the master file name from the user. The data acquisition system could perform this addition itself when data collection is finished. The system must then access the TSX/RT-11 files, and transfer them to the UNIX system in the appropriate format (i.e., read into the data base). If any important information is missing, the system should prompt for it.

SELECTION OF DATA FILES

The files to be used as input for an application program must be selected by the user. This selection process must be very flexible. The user may base his selection on criteria such as:

- the name of a raw or processed file,
- a combination of experiment and subject characteristics
  (subject name, weight, experiment type, etc.),
- a characteristic of the data in a processed data file.

Besides selecting data files, the user should be able to query the system for other information. For example, he may wish to see (or print out) the protocol for a particular

experiment. Or, he may wish to see what subjects have participated in an experiment.

The selection process must be easy and clear. The naive user should be guided through the process by a series of menus. A minimum of typing should be required, and the use of icons and a mouse is a desirable feature. The system should store the latest set of search criteria, and provide a hardcopy printout on demand. This will enable the user to repeat a search or to edit the criteria. For examples of easy and flexible search defining methods, see [19] or [20].

Information about the data files can be stored in a commercial data base. This information can be separated into tables (called relations) of information about different things (entities). For example, information about the subject is stored in one relation and details of the experiment in another. This format is called "relational" (see [7] for a description of different data base types).

The following structure for the data base is suggested. It will be possible to specify a value or range of values for most (if not all) of the items listed below. The Entity-Relationship (E-R) structure, showing the relations and the links between them, is given in Figure 2.

run : a relation containing specific information about each run of an experiment:

- subject name
- experiment ID
- unique run number
- date and time
- trial number and condition (eg. hypoxia, nitrous oxide exposure, etc.)
- drug or gas mixture (if it varies with each run)
- log_file name (and flag to indicate unusual event?)
- comments

subject : a relation containing detailed and relatively constant information about each subject who has ever performed an experiment. This information can be accessed by all experiments, and need not be collected every time a subject participates in an experiment. Thus there will be less duplication or contradiction of data. Eventually, the data acquisition system should be able to access this information whenever a subject's name is entered, so that the it can be checked and updated if necessary. This information will be protected against unauthorized access.

- subject name (full name is stored here, but will match abbreviations)
- SIN (Canadian or US)
- birth date
- sex
- race
- height
- weight
- handed-ness (right or left)
- phone number
- address
- drinking habbits
- caffeine consumption
- smoking habits
- exercise habits
- medication
- injuries
- medical history (diabetis, hypertension)
- comments

exper : a relation containing constant information about the experiment.

- experiment ID
- experiment name
- experiment code (eg. dive, rapid decompression, etc.)
- investigator's name
- gas mixture or drug used (if the same for all runs)
- protocol

Further experiment conditions, such as depth and bottom time for dives, are stored in the relations whose names correspond to the experiment code below. For example, the relation *dive* contains information specific to dive experiments.

raw_files : a relation containing a list of all raw files in the data base:

- unique run number
- raw_file name
- instrument code
- location
- part number (some data may be split among a series of files)
- comments

proc_files : similar to the raw_files relation:

- unique run number
- proc_file name
- processed file code (eg. for standard ECG statistics, etc.)
- location
- date
- comments (including details on options chosen when generating the file)

dive : this relation contains information characteristic of dive experiments. It will have an entry for each experiment whose code is "dive". There will be a relation like this one for each different experiment code.

- experiment ID
- dive type (linear, random)
- bottom time
- depth
- time to bottom
- stops
- decompression sickness

raw_proc : a relation storing the links between the raw and processed data files, so that the user may determine which came from which. The relation between the two types of files is "many to many": one processed file might be created from many raw files, and one raw file may be used to create many processed files.

Using this (or a similar) structure, a list of names and locations will be produced for the files which passed the selection phase. This list can be given (with comments describing the files content) to the user, or passed directly to the application modules. The user should be able to examine the selected files for comments, date, protocol, etc., to see if they are what is required. If not, changes in the search criteria can be made and the search repeated.

If the raw data files are not stored on-line, the system will have to provide the user with instructions on where to find the files and what to do to bring the files on-line.

The system should examine the log files for the selected experiments to see if any unusual events occurred. If so, the user should be notified of them before the data is analyzed.

It may become necessary to add fields to some of the relations. This function must be automated, so that the researcher need not interact with the data base itself. The addition of new relations will be necessary when a new type of experiment, with its own particular parameters, is developed.

APPLICATION

The set of possible application programs for data held in the PSMS is not limited. Although some applications (such as statistics packages) are likely to be fairly standard, smaller analysis programs are constantly being developed.

The user can choose any application which has been 'installed' in the system. When new applications are added, an interface between the system and the application will be written; this is the installation process. The interface will contain information about data format requirements (binary or ASCII, free form or fixed, header requirements, etc.). It will allow the PSMS to prepare files as inputs for the application. The installation of new applications should be as easy as possible, perhaps consisting of a series of questions and answers.

The user will not always want to send a complete file to the application program. He may want several files combined, or some information extracted from a file (for example, a sample every 10 sec). A header containing age, weight, height, etc. is often required. These values can be extracted automatically from the appropriate files in the data base. The PSMS should be able to insert codes to indicate missing data where this is the case.

The method of data restructuring must be easy to use, specific and unambiguous. A standard filter menu containing several standard features (e.g,. jumping over every so many values, or concatenating several files together) could be presented. Another possibility is a visual edit capability, so that the user can scroll through the data and select what he wants. The outputs from these operations can be sent directly to the application program, and stored as a processed data file so it can be used again (in this case, the *proc_files* and *raw_proc* relations would be updated). The program must be able to perform these operations on both raw and processed data files. Processed data files are often passed to a statistical package for analysis, while raw data files are usually sent to processing programs to extract important values.

After the format and application have been specified, the PSMS must perform the desired operations. The user must know exactly what is in the output files, so that he can create command files for the application programs. The structure of the data should be described (e.g,. 150 cases of four variables). This information could be incorporated into an IPAC-style header on the processed file itself, to be used by the application program, but the user must also be informed about the contents.

USER INTERFACE

There are many possible methods to enable the user to communicate with the PSMS. Some possibilities are menus, command languages, or voice. New ideas are being developed and existing methods improved.

The control structure for the PSMS must be easy to use by the researcher. The system should not require a long learning time, and it should be efficient and fast for the experienced user. The communication language must be accurate and unambiguous.

Because there are so many possibilities and differences between users, the choice of communication method should be flexible. If new methods are developed in the future, such as good voice communication or a 'natural language', it should be possible to add them to the PSMS. Eventually, when there are several working alternatives, individual users should be able to choose their own. Their preference could be stored in a 'profile' file, so that it is automatically selected when they use the system.

Some of the communication possibilities are discussed in the following paragraphs. Development of the PSMS will begin with one method only, and improvements and additions will be made as more about users' likes and dislikes are known.

A menu is a list of possible options the user has at any given stage in the system. Virtually all functions can be controlled through carefully written menus. The advantage of menus is that the user is not required to type, or to remember commands or syntax. Menus rely on recognition rather than recall vocabulary [9]. Naive users can be guided through a complicated process with detailed menus every step of the way.

For more experienced users, menus may prove frustrating. If the user knows the system well, he does not need to be told of all the possible choices, and may resent waiting while the menu is drawn. It is difficult to repeat a series of selections to perform a similar function several times. The path through the menus must be found each time. Some of the frustration that arises in a menu system can be relieved by allowing the user to select a 'level of verbosity'. Menus which the experienced user does not need can be suppressed, while the naive user can still be guided all the way.

Menus are sometimes combined with icons, small pictures symbolizing an object or function. The user moves the cursor (perhaps with a mouse) to the appropriate icon to perform a function. This method can remove almost all reliance on the keyboard. See [19] and [20] for descriptions of systems using this method to interface with a data base.

The user can control the system by a series of commands. For example, the user may request:
"run *dbpan* on raw force platform files where expt_code = dive and date > Jan 1985."
This is a realistic example since: *dbpan* produces processed files containing calculated parameters. See Annex B for an example.

A command language lets the user enter commands in the order he wants, and is more flexible than menus. Macros, which allow common series of commands to be summarized in one word, are possible. However, the user must be familiar with the available commands and the syntax required. Even experienced users may be unaware of some of the capabilities of the system.

Voice is a rather inefficient method of communication, in that more words are used in vocal communication than in any other method [6, 13]. Every speaker has a different way of speaking. Thus a vocal interface is very complicated to construct. However, as speaking is very natural for humans, such an interface may appear in the future. It will probably be similar to a spoken command language. When available, it could be incorporated into the PSMS.

E-R scrolling is a method of browsing through a data base suggested by Taylor et al. [19] for the Interactive Spatial Information System (see Annex A.). When the user wants to browse through the data base, an Entity-Relationship diagram of the data base will be displayed on the screen (see Figure 2 for an E-R diagram of the PSMS). The user moves the cursor to one of the relations (represented by rectangles) and presses a button. A submenu pops-up, listing all of the possible attributes of that relation. The user can pick and define any of the attributes he wants to. He can then move to another relation and set more attributes. It is always possible to return to a relation to change its settings.

This method provides a very natural and comfortable way of looking through the data base for the files wanted. It is rather like the data base query language QUERY BY EXAMPLE, which was shown to be more comfortable for users than command languages such as SEQUEL [9]. The concept of sub-menus could also be applied to application selection, and other interactions.

No matter which interaction method is chosen, it must allow repeated command sequences to be entered easily. The user should not have to re-enter all the criteria to perform a set of similar searches, or to analyze a series of files. Eventually, some combination of the interaction methods described above should be possible. For the time being, one will have to be selected for initial development of the system. On the basis of the above discussion, menus and icons should form the means of control. This combination is the easiest for inexperienced users. Some of the concepts described in the E-R Scrolling section could also be applied. Additional options can be added later.

Help functions are becoming much more common in systems today. They allow any user to request help at any specific stage in program operation. Inexperienced users can ask for help when they do not understand something, while experienced users can refresh their memory on some point they have forgotten. Help functions should be provided in the PSMS.

When designing the system, it is important to consider the response times required. The PSMS has two separate components: the search process, and the data restructure process. The search process in the PSMS is interactive, and should have low response times. The data restructure, once set up, does not necessarily require fast execution. The restructuring could go on in a batch fashion, in the background, so that the system can proceed with other work. The response times of the analysis itself will vary with the application program.

THE COMPLETE SYSTEM

The system environment is shown in Figure 3. The data files, raw and processed, form the core of the system. The actual storage method will be determined in the technical design phase.

The data files are surrounded by a data base interface. The data base stores information about the files, such as names and locations, origin, and contents. The data base allows the user to select files to analyze without having to remember their names. Instead, he can use facts he is much more likely to remember: experiment or subject name, or date, etc. The data base structure also allows analysis of similar groups (based on such things as height, weight, gas mixtures, etc.).

Surrounding the data base interface are a number of modules which allow the user to analyze the files he has selected. He can choose an analysis or application program to run on his data. To do this, the data may need to be restructured. The application interface stores particular information about the application program's requirements. The user may also want to browse through the data files, and perhaps graph them, to test their integrity. The input module adds new files into the system. Programmer access is given, to allow new application programs to be developed internally to fulfill specific needs. Other modules may be added in the future as new needs develop.

The outermost layer, controlling all the internal functions, is the user interface. This interface can take several forms, and eventually the user should be able to choose a favourite method. During development, the interaction is through a system of menus. Direct access to the data files and the application programs through the UNIX operating system is always possible.

FURTHER DEVELOPMENT

The system is feasible. Some technical matters will be settled during a feasibility study of the PSMS. Once this has taken place, the programming itself should be relatively fast.

Specific matters which must be investigated are:

- methods for performing the actual restructuring of data
- raw data file formats (IPAC, compressed format, etc.)
- a method of interfacing with ILS, which runs under RT-11
- logical organization of experiment and run characteristics
- interface design for adding new applications

It is estimated that an initial working version of the system could be delivered at a cost of approximately $50,000.00. Such a system would contain some of the proposed features but would take additional development work to meet the capabilities described in this document. The system is self-modifying, in that users can add applications and develop programs easily using the capabilities of the system. Thus, further

development can take place in-house, at small cost. Major additions, such as artificial intelligence capabilities, etc., can be implemented at a later time.

CONCLUSIONS AND RECOMMENDATIONS

There is a need for a system to aid researchers in the analysis of large quantities of physiological and other time series data. The proposed system will make analysis of complex physiological data easier, thus saving time and making more involved analyses feasible. A system which acts as a sophisticated interface between the data files and the application programs would fulfill this need.

Information about the files and the data they contain will be stored in a commercial data base. Access to the files themselves can be gained through the data base. Data files will be rearranged and combined to make them ready for different analysis programs.

A flexible user interface which guides the user through the selection of data files and application will make the use of the system very simple. Initial development will take place with a menu interaction system; other methods are possible in the future.

This report provides a conceptual discussion and design of the Physiological Signals Management System. It does not consider data structures, system architecture, or similar technical matters. It is recommended that, using this report as a basis, a study of these matters be undertaken. Following this study, programming could commence.

ACKNOWLEDGEMENTS

We wish to thank the assistance of Mr. Jim Tam, Mr. Dave Eastman, and Maj. G.Porlier in the preparation of this manuscript.

Figure 1. Typical physiological signals recorded on a strip chart

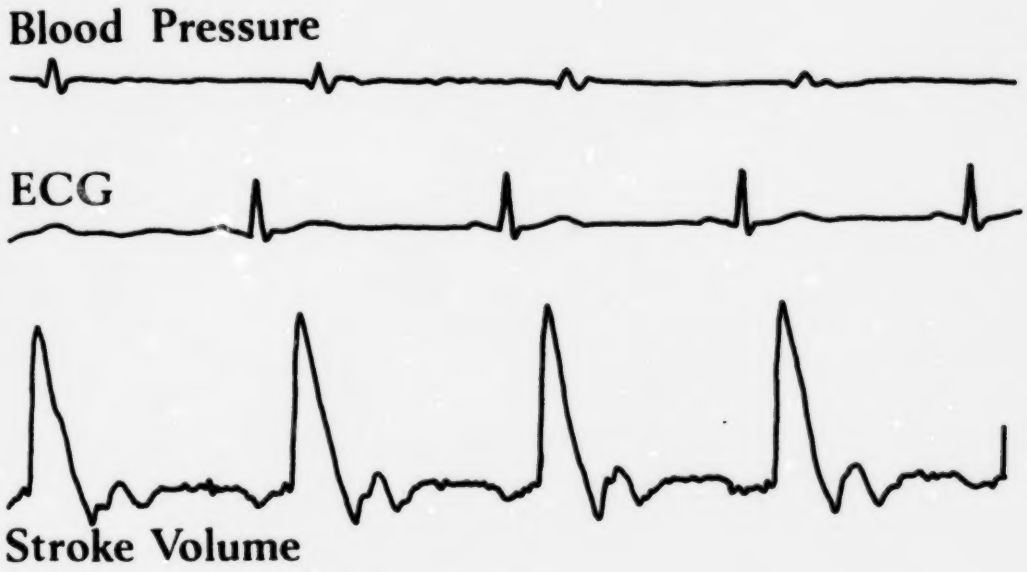


Figure 2. Entity-Relationship diagramn for the PSMS

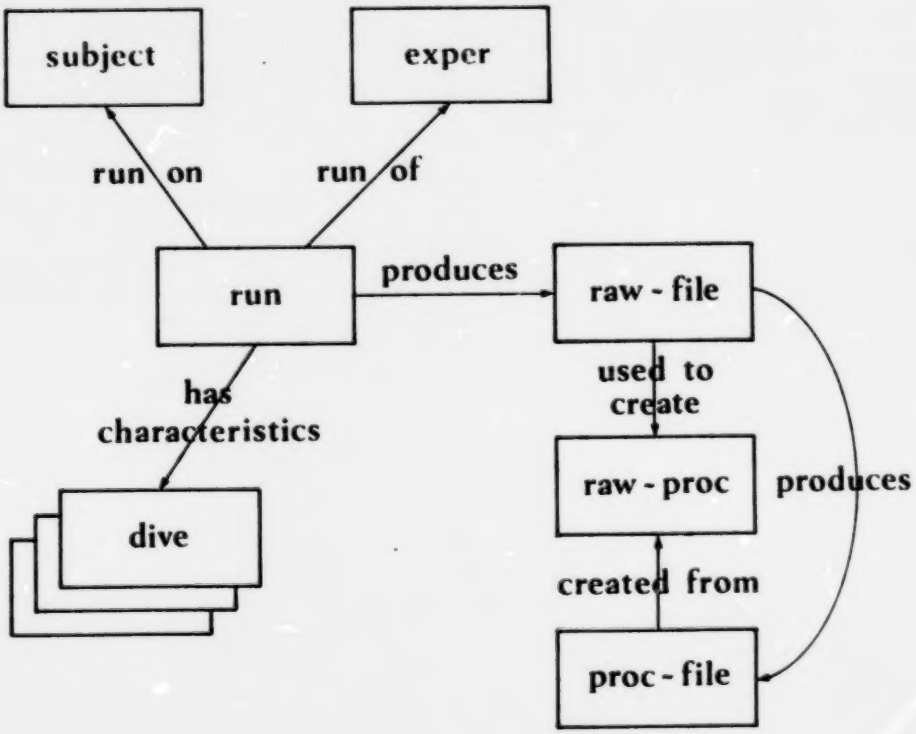
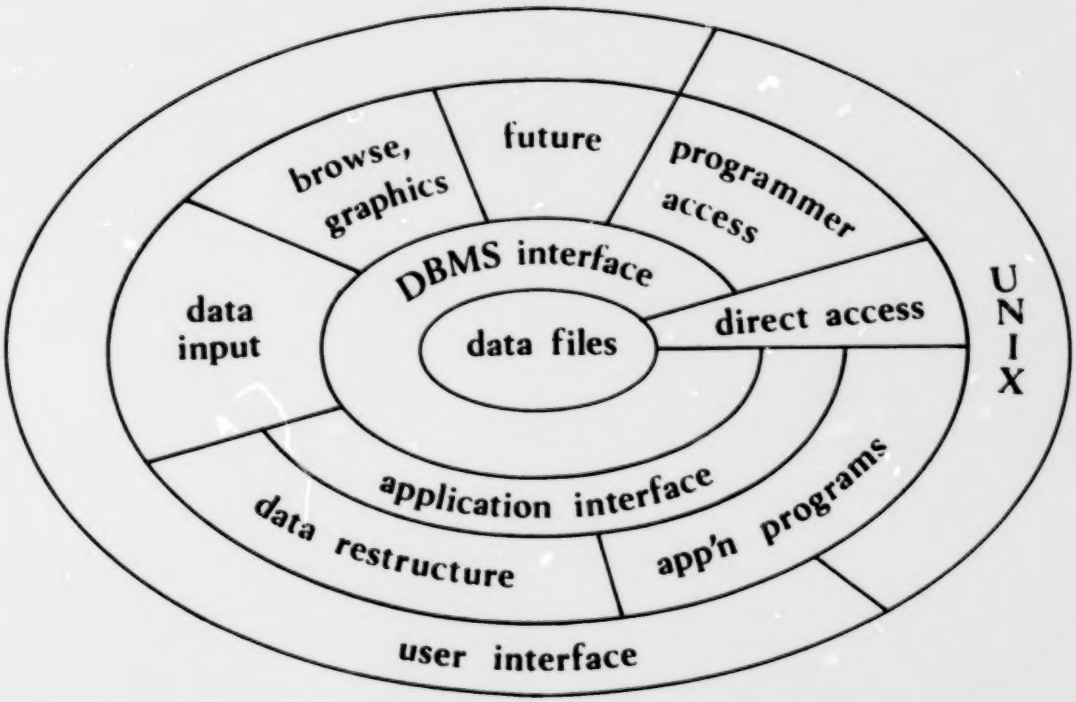


Figure 3. Overall system environment



Annex A

Spatial Data Base Management System (SDBMS)

DCIEM is investigating the computer-human interface for manipulating spatial information. As part of this investigation, an Interactive Spatial Information System (ISIS) is being designed. As the first phase of this project, a Spatial Database Management System (SDBMS) has been created [10-12, 19].

The ultimate goal of the project, the ISIS, will be an intelligent package capable of providing spatial information in the form most suitable for human analysis. The application envisaged is in military command and control, although other civilian applications exist.

The SDBMS-1 is a "small demonstration project to illustrate the value and practicality of managing spatial data in a slightly extended conventional database management system" [19, page 10]. It provides the capabilities to display and remove spatial information on a graphics terminal, modify stored information, and query the system about specific problems.

Some of the observations and conclusions about the user interface and the data structures are interesting, and may pertain to the design of the Physiological Signals Management System.

A special 'bulk' data type was added to a commercial relational data base (MISTRESS). This allows the inclusion of graphical data, such as coordinates. Spatial data is similar to signal data in magnitude, but differs in that it is reasonably well structured, whereas signal data can have any format.

The SDBMS is designed to accept any of keyboard, graphic, or voice input. Some combinations of these are also allowed. Commands can be entered as a sentence-like phrase, or by selecting the appropriate keywords from menus.

One interesting proposal is to allow the user to browse through the Entity-Relationship diagram for the data base (see Figure 2). The user can move to any one of the 'entities' (relations), and see a list of attributes of that entity he can define. Any combination of search criteria can be chosen, and the results of the search will be shown to the user so that he can change his selection criteria at will.

Annex B

Analysis of Force Platform Data

The force platform is a flat plate about 2 ft by 3 ft by 4 inches thick, which measures the displacement of the centre of gravity (postural stability) of a person standing on it. It is used as a measure of physiological trauma under various adverse conditions. For example, postural stability can be measured at high altitudes or at great depths, or while under the influence of drugs.

Signals from the force platform are collected using analogue to digital convertors, and stored by the acquisition program into a raw data file with a name such as 630622.p04. The raw data files are analyzed by another program, which produces a processed file containing about 100 calculated parameters describing postural stability. Combinations of these files are sent to BMDP for analysis. The proposed PSMS system would perform these operations quickly and efficiently.

The PSMS output file for force platform experiments is a collection of processed data from many (100) man-dives which are similar, taking place over a short period of time (eg. three months). For each run (a dive in this example), there is some header information (diver name, dive number, diver status, birth date, weight, height, etc.), followed by sets of complement blood enzyme data. Doppler bubble scores, and the force platform parameters for pre, post, and post-post dive. At present, the file is produced manually, that is, the header information is edited onto the force platform parameters using a computer screen editor, as is the complement and bubble data, and the files for each individual dive are strung together. Using the PSMS, this file would be created automatically as follows:

1. Select dives to be included in the file by going through the PSMS search process. For example, choose experiment equals force platform-dive, date between January and March 1985, and processed file code equals force platform parameters, complement values, or bubble scores.
2. Select the BMDP filter. This performs the following functions:

   For each selected dive,

   1. Determine header information (diver name, birth, weight, and height, status, dive number, decompression code).
   2. Determine if data exists for pre, post, and post-post dive force platform parameters, and create flags run1, run2, run3, accordingly.
   3. Add (append) complement values for the three periods to the header.
   4. Add bubble scores.
   5. Add the force platform parameters for the three periods.

   Combine all this information for all the dives. If any information is missing, fill the field with a 9.

References

1. A. ALLEN Ed., *SAS User's Guide: Basics*, SAS Institute Inc., Cary, N. Carolina, 1982.
2. *RS/1*, BBN Software Products Corporation.
3. BECKER, R. and CHAMBERS, J., *S: A Language and System for Data Analysis*, Bell Laboratories, 1981.
4. BROWN, M. B., ENGELMAN, L., FRAME, J. W., HILL, M. A., JENNRICH, R. I. and TOPOREK, J. D., *BMDP Statistical Software*, University of California Press, Berkeley , 1983.
5. BRUESTLE, H. R., SNIDER, D. M., RUSSELL, K. D. and STEWART, H. D., ISDMS - A Scientific Data Management System, *Software - Practice and Experience 15*, 5, (1985), 473-487.
6. CHAPANIS, A., Interactive Human Communication: Some Lessons Learned from Laboratory Experiments, in *Man-Computer Interaction: Human Factors Aspects of Computers & People*, B. SHACKEL (Ed.), Sijthoff & Noordhoff, Alphen aan den Rijn, The Netherlands, 1981, 65-114.
7. DATE, C. J., *An Introduction to Database Systems*, Addison Wesley, Reading, MA, second edition, 1977.
8. KUEHN, L. A. and SWEENEY, D. M. C., Canadian Diving Data: A Computerized Decompression Data Bank, *Computers and Biomedical Research 6*, (1973), 266-280.
9. MARTIN, T., Information Retrieval, in *Human Interaction with Computers*, H. T. SMITH and T. R. G. GREEN (Eds.), Academic Press, London, 1980, 161-175.
10. MCCANN, C., TAYLOR, M. and TUORI, M., *A Spatial Database Management System for Command and Control*, DCIEM No. 84-R-45, November, 1984.
11. MCCANN, C., TUORI, M., TAYLOR, M., MOON, G., TEASELL, S. and COTE, D., *Spatial Database Management System-1: User Interface Design*, DCIEM No. 84-C-13, March, 1984.
12. MOON, G., COLLINS, S. and BLAIR, G., Considerations for the Design of an Interactive Spatial Information System (ISIS), Final Report of Contract 8SE80-00052, 1984.
13. PALME, J., Interactive Software for Humans, in *Man-Computer Interaction: Human Factors Aspects of Computers & People*, B. SHACKEL (Ed.), Sijthoff & Noordhoff, Alphen aan den Rijn, The Netherlands, 1981, 167-213.
14. QUINT, S. R., GREENWOOD, R. S., HOWARD, JR., J. F. and GOMEZ, J. V., Automated data acquisition and analysis of neural evoked potentials, *Computer Methods and Programs in Biomedicine 20*, (1985), 35-44.
15. REEVES, B., *GPAC User Manual*, Computer Systems Research Group, University of Toronto, 1977.
16. ROSENTHAL, S. V., A Data Management System for the Collection and Analysis of Cardiovascular Analog Data, *Computers in Cardiology* , 1982, 421-424.
17. *Interactive Laboratory System V4.2 Training Manual*, Signal Technology, Inc, Goleta, California, 1983.

18. SILVERMAN, G., Automation in the Biomedical Laboratory, *IEEE Transactions on Biomedical Engineering BME-31*, 12, (December 1984), 748-752.

19. TAYLOR, M., MCCANN, C. and TUORI, M., *The Interactive Spatial Information System*, DCIEM No. 84-R-22, 1984.

20. WILLIAMS, M. D., What makes RABBIT run?, *Int. J. Man-Machine Studies 21*, 4, (1984), 333-352.

DEFENCE AND CIVIL INSTITUTE OF ENVIRONMENTAL MEDICINE
1133 Sheppard Ave West, PO Box 2000, Downsview, Ontario, Canada
Telephone (416) 635-2000